

# 第4章：関数（1 / 2）

春山 鉄源

2025/10/13, 10:22

## 1 関数とは

関数は一連の手順をまとめた便利なレシピのようなものです。関数を呼び出すと、Python はその指示に従って動作します。同じコードを何度も書く代わりに、関数の中にまとめておけば、必要なときに何度でも使うことができます。1. 組み込み関数 \* Python が起動すると、直ぐに使えるように様々な関数が用意されており、それらは組み込み関数（built-in functions）と呼ばれる。2. ユーザー定義の関数

## 2 組み込み関数

### 2.1 print() 関数

表示するための関数

関数名                      引数  
 $\underbrace{\text{print}}_{\text{関数名}}(\underbrace{\text{"GDP は 557 兆円"}}_{\text{引数}})$

```
print("GDP は 557 兆円")
```

コード 4.2.1

```
print("GDP は\n557 兆円")
```

f 文字列 \* 文字列の前に f を書く \* 文字列の中で {} を使う \* 割り当てた変数の値や計算結果などを表示できる。

```
x = 2 / 3
print(f"x の値は{x}")
```

x の値は 0.6666666666666666

```
print(f"x の値は{x:.3f}")
```

x の値は 0.667

<:.3f の解釈> \* : はこの後に続くコードは表示に関するものだと宣言している。\* . は小数点表示に関しての設定であることを示している。\* 3 は小数点第 3 位を示している。\* f は float の f（省略可）

3f を 5f にすると、小数点第 5 位までの四捨五入となる。試してみよう。

## 2.2 関数の呼び出しと実行

授業では扱いません。

## 2.3 type() 関数

授業では扱わない。

コード 4.2.3

```
x = 3.14
type(x)
```

### 2.3.1 例 4.1

```
type(10)
```

```
type(0.5)
```

```
type("GDP")
```

```
type([1,2])
```

```
type((1,2))
```

```
type({"GDP":592})
```

```
type(True)
```

## 2.4 さまざまな関数

教科書の表 4.1 には色々な関数を紹介しているが、ここでは次の関数を説明する。

### 2.4.1 sum()

合計を返す関数

```
x = [1, 2, 3, 4, 5, 6]

sum(x)
```

21

### 2.4.2 len()

要素数を返す関数

```
len(x)
```

6

x の平均は次のように計算できる。

```
sum(x) / len(x)
```

3.5

#### 2.4.3 abs()

絶対値を返す関数

```
abs(-10)
```

10

#### 2.4.4 range()

等差数列のオブジェクトを用意する関数

```
range(start, stop, step)
```

- start: 最初の整数（デフォルト、引数を与えない場合は 0）
- stop: 最後の整数の次の値
- step: 隣接する整数の差（公差）（デフォルト、引数を与えない場合は 1）

例えば、0 から 9 までの 10 の整数を準備するには

```
range(10)
```

```
range(0, 10)
```

↑↑↑ 0 から 10 までの整数が表示されないが、使えるように裏で準備されている。  
次のコードは 0 から 999,999,999（10 億-1）までの 10 億個の整数を準備している。

```
range(1_000_000_000)
```

```
range(0, 1000000000)
```

- 遅延評価: 実際に使うときに、0、1、2、3、、と順番に整数の生成が実行されことになる。
- for ループではよく使う関数となる。

#### 2.4.5 list()

リストを作成する関数

```
z = range(10)
```

```
list(z)
```

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

`range(10)` で準備された 0 から 9 までの整数を使って、リストを作成している。

次のコードをアンコメントして（#を削除して）実行すると 10 億個の整数からなるリストを作成することができる。しかし、終了まで数分掛かるだろう！ You are warned!

```
# list( range(1_000_000_000) )
```

実行を中止する場合 \* メニューから「Kernel」→「Interrupt Kernel」\* メニューから「カーネル」→「カーネルの中断」